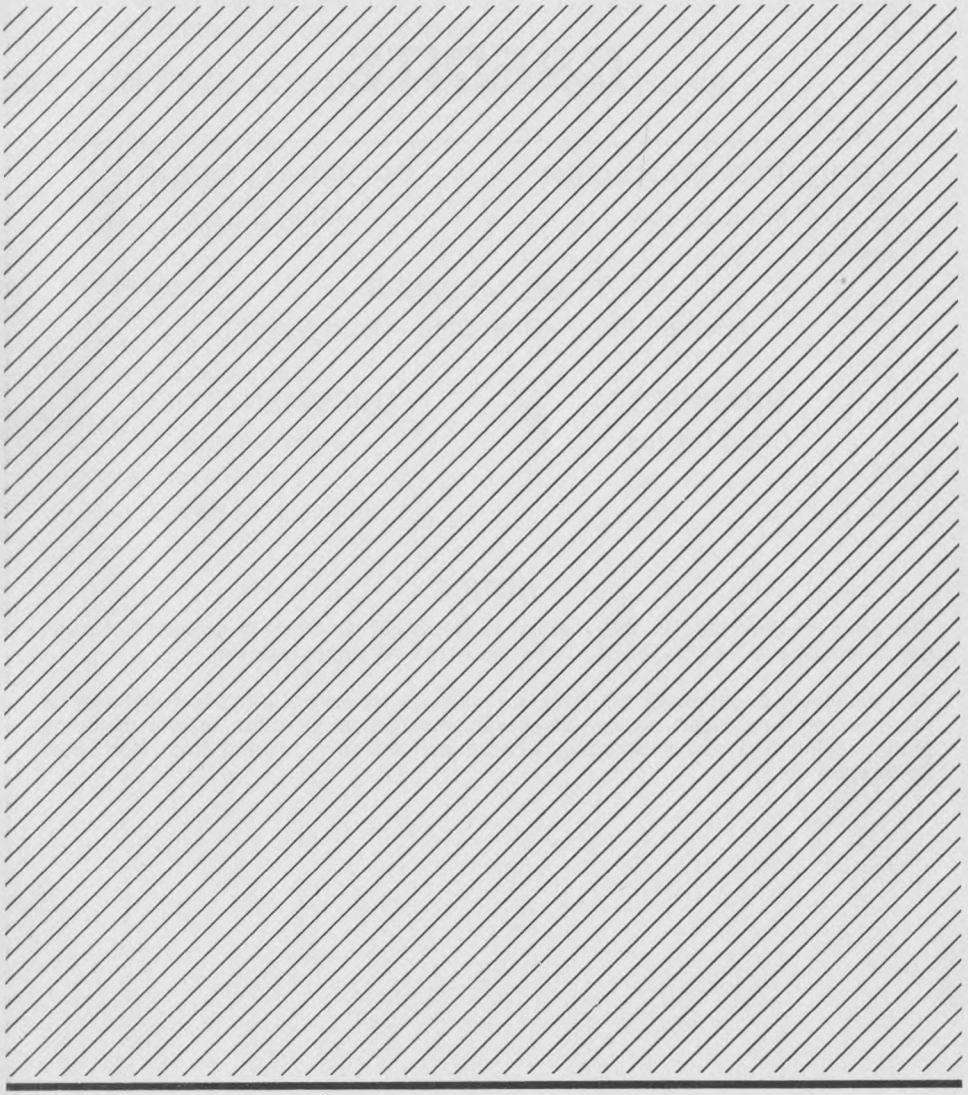
TRAINING KIT

NORTHWEST SOFTWARE



CREDITS

PROJECT MANAGER
Ming S. Low

PROGRAM DESIGN
Ming S. Low

PROGRAMMERS
Michael Fong—Sinh Siha—Jean Nevotti

GRAPHICS
Craig Davies

DOCUMENTATION
Jean Nevotti

NOTICE

The program described in this manual is Copyright © 1984, Northwest Software, 621 Minna Street, San Francisco, California 94103. All rights reserved.

This manual is Copyrighted, all rights are reserved. No part of this document may be reprinted or reproduced by any electronic, mechanical or other means including photocopying and audio recording without permission in writing from Northwest Software Corp.

TABLE OF CONTENTS

SECTION ONE: DEFINING TERMS

The Computer as an Input—Output Tool
A Conceptual Overview 2

Using the Computer as a Tool
Modes of Operation 4
The Problem of Communication 6

SECTION TWO: COMMODORE 64 OPERATIONS

Getting Familiar
The Keyboard 8
Memory Concepts 10
Data Concepts 11

The System
BASIC Concepts 12
Graphics Concepts 13
Sound Concepts 18

APPENDIX:

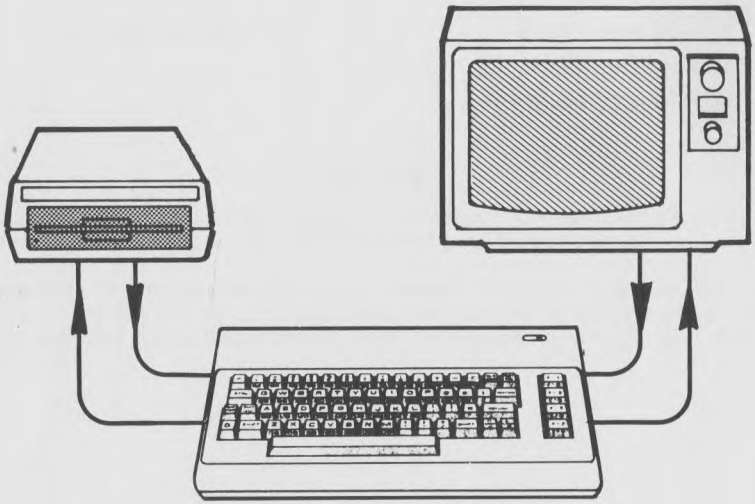
Crossword Puzzle 22
Recommended Reading 25

THE COMPUTER AS AN INPUT/OUTPUT TOOL:
A Conceptual Overview

Most likely you have a good idea of the wonderful things that a computer can do. You'll probably be surprised to hear us call your computer a "dumb" machine. What we mean by this is that your computer cannot do anything until it is instructed what to do in a "language" that it understands.

Think of your computer as a very complex *adaptive* tool. What we mean by *adaptive* is that although it really only does one thing, manipulate data, it can do this in many different ways. Whether you are using the computer as a word processor, a game machine or a teacher it is doing the same thing: following written instructions. These intructions are almost always written in the computer's primary instruction set, *the BASIC language*, and are input through the keyboard.

Let's take this idea one step further. We've established that the computer will only respond to instructions or *input*. We also know that this *input* must be in a language that the computer understands, usually the BASIC language. Obviously, if we are going to the trouble of instructing the computer we expect some kind of reaction or *output*.



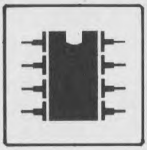
Output Options

Your Commodore 64 has many *output* options. These are grouped into four fundamental categories:

• SCREEN OUTPUT: This is what the computer is capable of reporting to your monitor. It can produce **text** (letters), **symbols, graphics** and **colors**. In other words, your Commodore 64 is capable of producing all of the standard visual elements of perception on the screen. It can even simulate depth through the creative use of color graphics!

• HARD-COPY OUTPUT: This is what the computer is capable of sending to your printer or plotter. (By *hard-copy* we mean "on-paper".) Your computer can send anything to a printer that it can send to a monitor. It is the limitations of your printer or plotter that dictate what will appear on the paper. Most printers only print those letters, numbers and symbols that appear on a standard typewriter. Some printers also print "character graphics". More sophisticated printers and plotters will produce just about anything you can see on your screen including full graphics and colors.

• AUDIO OUTPUT: Your Commodore 64 can also produce a wide range of sounds. These sounds can be controlled as to pitch, frequency, duration, octave, etc. Some experienced programmers have even succeeded in making the Commodore 64 **output** human voice simulations using this audio output feature.

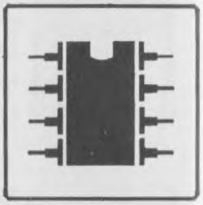


• DEVICE OUTPUT: Your Commodore 64 is capable of sending its output to other electronic devices. These include other computers (through a *modem*), and data storage units (disk drives and cassette recorders). Many other electronic devices have been designed to understand Commodore 64 output. These vary from complex medical intruments to simple switches which can turn your coffee pot on in the morning.

It must be understood that to get the desired output you must give exacting input which tells the computer WHAT to output, HOW to output it, WHEN to output it (in relation to the rest of the output) and WHERE to output it. Without all of these intructions your computer will either give its output incorrectly or will do nothing at all. The process of instructing your computer what, where, when and how to output is called ***PROGRAMMING***.

USING THE COMPUTER AS A TOOL:
Modes of Operation

Like many modern tools, the computer can be applied to a wide variety of different uses. Although there are hundreds of these potential *applications*, below is a list of the most common ones.



PRODUCTIVITY: The computer can make us more productive by storing data (such as mailing lists), manipulating written data (word processing), calculating (everything from taxes to statistical data) and graphing (interpreting complex numerical data into visual symbols which we can easily understand).

ENTERTAINMENT: Because of the computer's unprecedented complexity, it is the ultimate game player which can challenge everything from our sensory abilities to our intellectual prowess.

EDUCATION: The computer's ability to present ideas using a wide range of methods (audio, graphic, text) and the fact that it is tireless makes it a capable teacher.

COMPUTER LITERACY: Many people use computers as a tool to learn about computing. With the advent of computer use in our society, many of those who have no interest in using the computer for more practical purposes feel a need to know more about them.

Remember that your computer can't do any of these things without the proper instructions. With this in mind, lets talk about methods of instructing the computer or *Modes of Operation*.

THE FOUR MODES OF OPERATION:

• RUN-TIME MODE: Most people who have complex tasks such as those listed above purchase pre-programmed software (such as this Training Kit) because they have neither the expertise nor the time to write the instructions necessary to do those tasks well. Such users spend the majority of their time in run-time mode. This mode is characterized by complete program control. When you are in *run-time mode* all interaction that you have with the computer is actually interaction with the program that is running (therefore the term "run-time"). You do not "program" in run-time mode, you simply operate the program that is running.

• IMMEDIATE MODE: When you turn your computer on it is in immediate mode. From this mode you can do a limited number of things. Any instructions that you give the computer will be excecuted immediately and then forgotten. (Therefore the term "immediate mode".) No immediate mode operations will be remembered by the computer. Immediate mode is generally used as an avenue to some other mode of operation. The command LOAD which gets a program from a storage device, and the subsequent command RUN which tells the computer to enter *run-time* mode are both input in immediate mode. When in immediate mode the computer can also be used as a simple calculator.

• PROGRAM MODE: When writing a list of instructions or a program, you are in *program* mode. When in program mode, the computer will accept almost any written instructions (even incorrect ones) and allow you to continue in your list. The computer does not analyze or execute those instructions until it is told to enter *run-time mode*. The instructions used in program mode are the same as those used in immediate mode, except that each instruction line starts with a number between 0 and 63999. This number tells the computer not to excecute that instruction immediately but instead to save it for later execution.

• EDIT MODE: Edit mode and program mode go hand in hand. Edit mode is that mode in which instructions (or *programs*) are enhanced in some way. Through edit mode a program can be lengthened, shortened or altered.

Editor's Note: One can switch from immediate to program to edit mode with as little as one keystroke. For this reason it is traditional to refer to them collectively as "Program Mode". We have created these categories because we felt that this was a logical distinction for beginners. As you become more experienced you may not find it necessary to make this distinction.

USING THE COMPUTER AS A TOOL:
The Problem of Communication

Without instruction, the computer will do nothing. Even the user who only intends to use pre-programmed software must know enough instructions to tell the computer to load the desired program and enter run-time mode.

To this end, your Commodore 64 was provided with an *instruction set*. This instruction set is called BASIC which is an acronym for:

Beginners **A**ll-purpose **S**ymbolic **I**nstruction **C**ode

BASIC provides you with 65 words with which to command your computer. These commands in combination with the correct *values* and *syntax* allow you to instruct your computer. Let's look at an example:

The LOAD command is one of the most common BASIC words used. It tells the computer that you wish to load a program into the computer's memory. It does not tell the computer which program you wish to load or which *device* (disk drive or cassette) that you wish to load it from. Therefore, we must add some "values" to the word "load" which make the command more specific.

To tell the computer that we wished to load the Training Kit program, we would type:

```
LOAD "TRAINING KIT"
```

Note the quotation marks. These are a part of the syntax that the computer uses to interpret the instructions we give it. Without the quotation marks the computer would not be able to distinguish between the command (LOAD) and the value (TRAINING KIT).

If we were to type in this command as it is above, the computer would assume that we were using a cassette and would attempt to find the program on the cassette recorder (datasette). If we happen to be using a disk drive we must tell the computer that we are using a disk drive within the instruction. The command would then look like this:

```
LOAD"TRAINING KIT",8
```

The comma is another part of the command syntax. In combination with the LOAD command, it tells the computer that the value which follows it refers to the storage unit involved. The "8" is just the computer's way of saying "disk drive" so we don't have to type it out.

If we are to properly instruct the computer we must learn the command vocabulary and how it relates to our computer use. With each command we must also learn the syntax involved with its use. The Training Kit program introduces this process in a logical, easy-to-understand way.

REVIEW:

Below are some of the key concepts that we've learned:

1. The computer is an input/output tool. Information is generally input through the keyboard. Output is generally either visual (through the screen), audio (through a speaker), hard-copy (through a printer) or given through some other device designed to understand information from the Commodore.

2. The process of telling the computer what, when, where and how to output information is called programming. This is usually done in the BASIC language.

3. The Commodore 64 can play games, educate or be used in many other ways. These uses are called applications.

4. Most users spend the majority of their computing time in RUN-TIME mode. One does not program in this mode; one operates a program that is running.

5. The BASIC language is an instruction set. It contains 65 words which are used to tell the computer what to do.

THE KEYBOARD

All direct communication with the computer is done through the keyboard. With only 66 keys, the C-64 keyboard can generate 255 separate characters and symbols in any of the 16 colors. Some key combinations will even cause the computer to perform simple operations in as few as two keystrokes. The ability to access these capabilities is an important part of the process of computing.

THE STANDARD CHARACTER SET

This is the most commonplace character set and includes the alphabet, numbers and standard syntactic symbols. It is the easiest character set to access. To use this set of characters, simply type as if on a typewriter.

KEY COMBINATIONS

The rest of the characters including reverse letters and the graphic set (the symbols on the front of the keys) must be accessed through the use of key combinations. Almost every key on the keyboard will respond differently when depressed in combination with certain important keys.

KEYBOARD MODES

When the Commodore 64 is turned on it is in *Upper Case/Graphics* mode. This means that any letter you type will appear as a capital letter and when you press the key in combination with the SHIFT key it will produce a graphic symbol.

There are only two modes of operation. In the other, *Lower Case/Upper Case* mode, the keys will produce lower case letters and capitals when pressed in combination with the SHIFT key. This mode is much like a typewriter.

To switch from mode to mode simply hold down the SHIFT key and press the Commodore key.

COLORS

The Commodore 64 is capable of producing 16 colors on its screen. These colors are usually produced using BASIC commands but may be produced directly from the keyboard. To type in color simply hold down the CONTROL key and depress any number between one and eight. You will have produced the color labeled on the front of the number key selected. To produce the remaining eight colors hold down the COMMODORE key and press any number between one and eight.

GRAPHIC SYMBOLS

On the front of each alphabetic key you will see two graphic symbols. These symbols were designed by experts in the field of perception and can be used in combination with one another to create various types of drawings and illustrations. It is easiest to access these keys through UPPER CASE/GRAPHICS mode. To access the character on the right hold down the SHIFT key while depressing the desired symbol's key. To generate the character on the left hold down the Commodore key while depressing the desired symbol's key.

IMPORTANT KEYS:

1. RETURN
Use this key when done inputting any one command or program line. Tells the computer that you are ready to proceed.

2. SHIFT
This is the most commonly used combination key. It will cause almost every other key to respond differently than it normally would. Will do nothing when depressed alone.

3. SHIFT/LOCK
The equivalent of SHIFT continually depressed.

4. CONTROL
Used mainly to produce colors and reverse characters, the CONTROL key is also used by many programs for specific functions. Will do nothing when depressed alone.

5. COMMODORE
Another combination key that will do nothing when depressed alone. Mostly used in combination with shift to switch from lower/case to upper case mode and to produce colors.

6. CURSOR KEYS
Cause the cursor to move on the screen both horizontally and vertically. Move cursor down and left unshifted, upward and right shifted.

7. RUN STOP
An *operations* key. Causes the computer to "break" or stop a program that is running. Shifted inputs cassette LOAD command.

8. RESTORE
Another *operations* key. Mostly works in combination with RUN/STOP key to return to immediate mode when desired. Resets computer memory but does not erase it.

9. INSERT/DELETE
Deletes the character left unshifted. Inserts a space right when shifted.

10. CLEAR/HOME
Puts the cursor in the home position (upper left corner) unshifted. Does this and clears screen shifted.

11. THE FUNCTION KEYS
The function keys are user programmable keys that can be used to generate symbols, interact with programs or input complex operations. Let's say you spent all day loading and saving programs. You could program two of the function keys to type in the words "load" and "save". Thus, you could press F1 instead of typing "load" and save keystrokes.

KEY COMBINATIONS

RUN/STOP & RESTORE

This combination causes the computer to stop what it's doing, return to immediate mode, clear the screen and keep the program in memory, all in two keystrokes! Hold down the RUN/STOP key and *tap* the RESTORE key. (The RESTORE is the only key that needs a sharp tap to work.)

CONTROL & 9 or 0

Hold down the control key and press the number 9 key to display characters in reverse. Hold down the control key and press the number 0 key to return to normal display.

SHIFT & CLR/HOME

Causes the screen to clear and the cursor to return to the upper left hand corner of the screen.

SHIFT & ALL FUNCTION KEYS

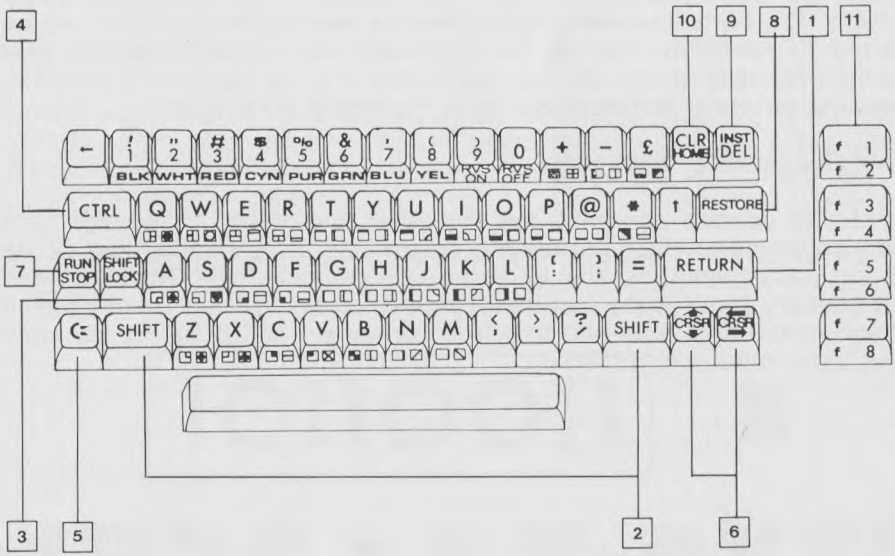
You'll note that there are only four actual function key caps; to produce the remaining four function keys, depress the desired key in combination with SHIFT.

SHIFT & CURSOR KEYS

Shifted, the vertical cursor key will move up. The horizontal cursor key will move left when shifted.

COMMAND FUNCTIONS

Fifty two of the BASIC commands can be abbreviated using SHIFT key combinations. See page 130 of your Commodore 64 Users Guide.



MEMORY CONCEPTS

Computer memory is a collection of cells, each of which is capable of holding a fixed amount of data. This data either represents information or it instructs the computer what to do, depending on the context in which it occurs.

Think of memory as a bank of post office boxes in a large city post office. (The Commodore 64 would have 65536 boxes.) Each box has a box number or address.

Our "mail" is data. Data can be instructions, values or any information we would like to see "delivered" or processed. Most computers have two different kinds of memory. Let's look at how these memory types relate to normal computer use.

RAM (Random Access Memory)

To use the computer we must be able to store data and instruct the computer how to use that data. These data and instructions are placed in the computer's memory so that they can be processed. This memory is known as RAM. The term *Random Access Memory* means that the data can be referred to randomly; that is, the computer can instantly access data anywhere in its memory. You can store data in or access data from RAM. When the computer is turned off all of the data in RAM is lost.

ROM (Read Only Memory)

In order to operate correctly, the computer must have a language with which to interpret our instructions as well as an understanding of its features and capabilities. This information is stored in ROM. The term *Read Only Memory* conveys the fact that ROM cannot be "written" to or used as storage space by a user and that it is permanent and does not go away when the computer is turned off.

THE DETAILS OF DATA

At the risk of being aggravating, we'll repeat that the computer's only real function is to manipulate data. *Data is the form that information takes when being computed.* It is insignificant when taken out of context just as a letter is insignificant when seperated from a word or a word from a sentence.

In order to efficiently process data, the computer breaks it down into its smallest possible unit. This unit is called a bit.

BIT

Because the computer thinks in binary terms, a bit is always either the number one or the number zero.

BYTE

Bits are grouped into sets of eight. Each group of eight ones or zeros represents a character (number, letter or symbol). This character is referred to as a byte (pronounced "bite"). The word "walk" is a four-byte word made up of 38 bits.

K-BYTE ("K")

Surely you've heard the Commodore 64 referred to as a "64 K" computer. This is a term adopted by computer users to describe 1000 bytes. (Taken from the metric word "kilo" or thousand.) Though it is not important, 1 K is actually equal to 1024 bytes.

We encourage you to integrate these divisions of data into your vocabulary.

BASIC NOTES:

This section is not designed to teach you how to program in BASIC; rather, we wish to discuss some fundamental BASIC concepts and lay the foundation for a good understanding of BASIC logic.

There are many excellent books devoted to BASIC programming on the Commodore 64. See the references at the back of this guide for more information.

USES OF BASIC:

BASIC can be used on many different levels. Because it is the Commodore's primary instruction set all users should know at least a command or two. For those users who have no desire to program it is not necessary to spend much time researching or studying BASIC. For others it can be a science which takes months to master. Commodore BASIC is not so complex that it would take much more than a couple of months to know thoroughly.

On the most fundamental level BASIC is used only to access pre-programmed software from storage. The BASIC keywords that handle these procedures are called *system commands.* These are the most widely used keywords and are easy to become familiar with.

BASIC can also be used to get immediate output. The key-words used for these operations are called *immediate mode commands* and operators. With these commands you can control screen output and do calculations.

BASIC is most complex when being used in the form of a program. The greater the sophistication of the program, the greater the difficulty and complexity of BASIC.

HISTORY OF BASIC:

In understanding BASIC and its relationship to your use of the computer, it is helpful to know its origins.

The first computers were instructed directly in their own language. This language, as we mentioned earlier, consists of a series of ones and zeros, each with significance as a result of their relationships. The task of instructing such machines was enormous and took both patience and extensive training. As the applications to which computers were applied became more streamlined, certain operations and commands became standard and were replaced with condensed abbreviations. This saved programmers time and aggravation. These abbreviations and symbols became the first computer "languages".

The problem with such languages is that as they became more complex the task of programming became more difficult to understand; a programmer had to know both the logic of the computer and the logic of the language.

It was in the early 1960s that two college professors faced with the task of teaching programming decided to develop a simplified language that used the same linguistic logic that the English language uses; in this way the new programmer need only know computer logic to program effectively.

Originally the language was only designed for educational purposes but was later adopted by manufacturers and evolved into the most popular computer language in the world.

LIMITATIONS OF BASIC:

Why, you might ask, if BASIC is so great and so simple, is it not the only computer language used? There are two major reasons. The first is that BASIC must be interpreted by the computer which takes time, so much time that BASIC programs are generally much slower than programs written in other languages. Also, because BASIC is really just a group of symbols, a programmer can only use as many of the computer's capabilities as there are corresponding symbols. In spite of the fact that BASIC is relatively well developed, there are more functions than there are accompanying symbols.

GETTING TO KNOW BASIC:

We believe that the best way to learn BASIC is to use it. Start with a simple application, perfect it and continue. A good BASIC reference guide is also extremely helpful if you are willing to invest $15.00 or so.

PROGRAMMING IN BASIC

The computer will only store your instructions in its memory if you are in *program mode*. To enter this mode type a line number (any number between 0 and 63999) at the beginning of your instruction. The computer will then know that what follows are instructions to be stored but not excecuted until it is told to *RUN* the program.

To instruct the computer using BASIC keywords in *immediate mode,* simply type your instructions and press the RETURN key. They will be excecuted and forgotten.

You cannot hurt your computer in any way using BASIC. If you experience difficulties, try the RUN/STOP & RESTORE key combination. If the computer reports an "error" (usually a 'syntax error') you entered your instructions incorrectly. Try again.

GRAPHICS:

Surely you've heard all the talk about the Commodore 64's graphics capability. We feel that C-64 has the finest graphics of any machine in its class.

Many practically minded people wonder "why graphics"? What's the big deal?

THE VALUE OF GRAPHICS:

Computers are used in many different ways for an incredibly wide range of functions. One of the beauties of the personal computer is that it offers a new medium for communication of ideas. The old cliche "A picture is worth a thousand words" is often true. The computer is the first device that could almost instantly convert "a thousand words" (or values) into a picture. A computer is also the first medium that could simultaneously present visual, audio and literal ideas (like a movie) and at the same time be fully interactive.

Furthermore, for those who are professionally involved with visual ideas (architects, designers, etc.), the computer offers a clean canvas, a full color palette and a straight edge in one integrated package.

Computer graphics is in it's infancy now, but will become an integral part of education, entertainment, business, commercial art and fashion as technological advances allow. The day may come when business letters will be sent in full color with appropriate illustrations included. Complex symbols may soon become as much a part of our written vocabulary as the alphabet is today.

As an average user you may wish to use your computer's graphic capabilities to make non-graphic tasks easier to understand. (Important items in red, arrows as pointers, etc.) Let's examine the most common types of Commodore 64 graphics in greater detail and discuss some of their common applications.

COLORS

The easiest and most accessible graphic elements are colors. The Commodore 64 provides 16 colors. You may display all sixteen colors on the screen simultaneously if you wish. It should be noted that many combinations don't give very clear output. For a table of workable and unworkable color combinations see page 152 of the Commodore 64 Programmers Reference Guide.

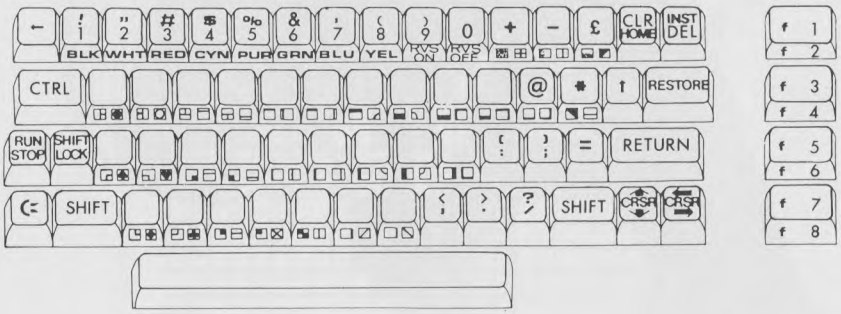
When programming in color you may select the color of the border, the background and the characters. Use colors that offer high contrasts but are aesthetically pleasing. If your program is likely to be used on a television set or monochrome monitor use only those combinations listed as excellent in the aformentioned reference guide.

TEXT GRAPHICS:

The shapes which are printed on the front of your keys are called text graphics. They work exactly like the other characters do. As opposed to using letters to form words, one uses text graphics to form images.
Text graphics are very simple. A patient programmer can create large letters, complex structures and even stick figures with these characters. (Some of the screens in the Training Kit were designed with the use of these characters.) Experiment with this feature by drawing simple shapes on the screen.

Text graphics are most often used to draw boxes around significant areas of the screen, to underline and to point.

(Note: Some text graphics are used in BASIC as "tokens". These tokens represent BASIC commands. For example, the text graphic of a heart shape is the token for the BASIC command "clear screen".)



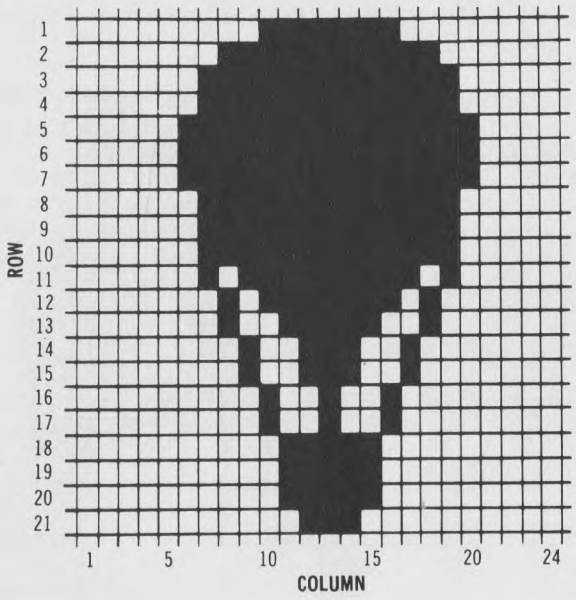
SPRITES:

The most talked about Commodore 64 graphics, sprites are images that you can program. These images are very attractive to programmers because they can be any shape and can move anywhere on the screen without disrupting the rest of the screen. Sprites can be no larger than about two square inches. Without special programming there can be no more than eight sprites on the screen at a time. Sprites can be either one or two colors.

Sprites are often used for animation. In a now-famous Commodore demo program, a character named "Clyde" walked out on the screen. Clyde was a cluster of sprites; one for each leg, one for the body, one for the head. Each sprite was a different color and moved independently of the whole.

There is a whole world of potential problems that sprites could cause. For instance, if a sprite graphic character is walking down a text graphic street, which should receive visual priority: the sprite or the street? What happens when sprites "collide" on the screen? Such questions were considered by the designers of the sprite feature. All of this information can be easily programmed.

On a practical level sprites are primarily used to animate characters or objects. There are over 20 programs now on the market to help you build, animate and understand sprites. These are called sprite editors and are remarkably helpful if you use this feature often.



HIGH RESOLUTION GRAPHICS:

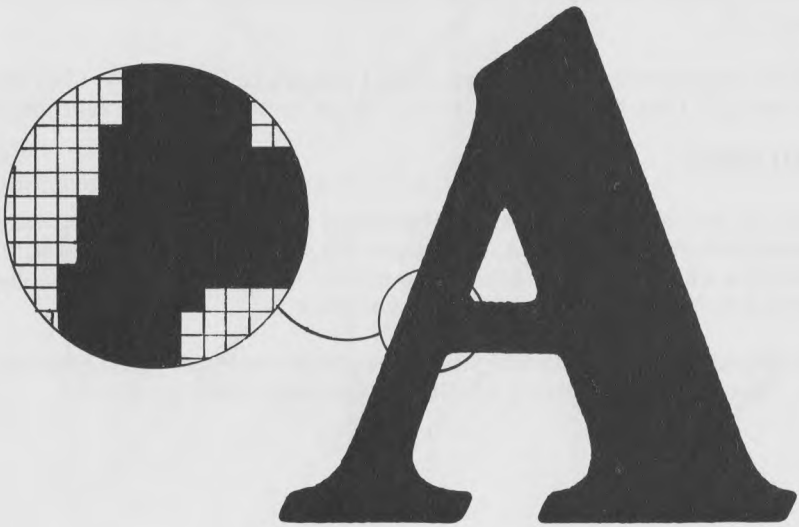
The most complex form of graphics, high-res graphics allow you to define your own characters, shapes and symbols.

Your computer screen, like a television tube, is a grid of tens of thousands of tightly packed dots. These dots are called *pixels* or "picture elements". Programming high-res graphics is the process of telling each dot to turn either off or on. In high resolution mode you can use each dot (64000 in all) but are only allowed one color.

A secondary mode, medium resolution, is much like high res in that each dot is individually programmed, except that in this mode pixels are grouped into sets of two which leaves you with 32000 units and therefore less resolution.

High and medium resolution graphics can be used for just about any image. The greatest advantage to these graphic types is complete creative freedom coupled with complete control. For example, high-res mode is the only graphics mode in which a fairly clean circle can be programmed.

Because high-res and medium-res modes give the computer a single instruction for each pixel, the computer must process each instruction individually. Therefore it is inappropriate to program in these modes within the already slow BASIC instruction set. Generally, high and medium resolution graphics are programmed in the machine's own language (otherwise known as *"machine language"*).



SOUND AND THE SOUND INTERFACE DEVICE (SID)

The Commodore 64 comes equipped with a sound generator called the Sound Interface Device or SID. The sound feature has been used to generate everything from simple tones to complex music. It is even possible to program SID in such a way that it will simulate the human voice!

Before you are able to effectively program sound, a knowledge of the principles of sound is helpful. The Training Kit program will concentrate on how to make sound with the SID. This section will focus on sound itself as it relates to the SID.

WAVES:

Sound is created by vibration. The life of a vibration, from the time it begins to the time it dies is called a *wave.* You hear sound when a sound wave travels to your ear. Controlling sound is done by instructing or *defining* sound waves.

PITCH:

The length of a wave from peak to peak is called a wavelength. The shorter the wavelength, the higher the pitch of the sound.

A common way to measure the pitch of a wave is in *hertz.* A wave which has taken one second to go from peak to peak is a "one hertz" wave. A wave that vibrated 200 times per second would be said to have a *frequency* of 200 hertz. The higher the frequency, the higher the pitch.

The average person can hear frequencies between 20 and 20,000 hertz. A 20-hertz sound would be dull and very low. A 20,000 hertz sound is shrill and very high.

When programming the SID you must always tell it exactly what frequency is desired. This is done by giving a value to one of SID's *tone oscillators.*

VOLUME:

Just as the length of a wave determines pitch, the size of the wave determines volume or loudness. The larger the wave, the greater the volume. The size of a wave in no way affects its pitch. Two sounds can have exactly the same pitch and yet have different volumes.

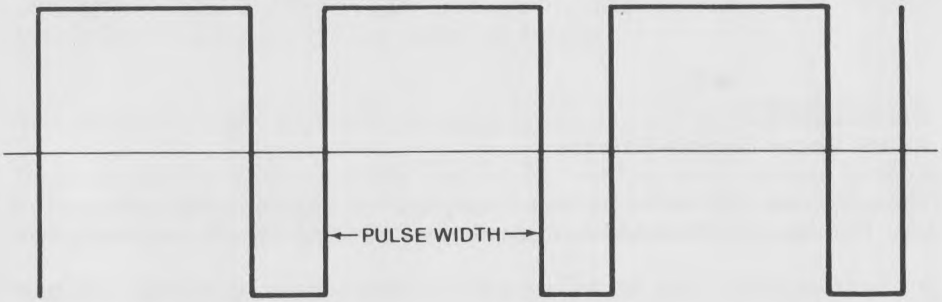
When programming the SID you must tell it how large you'd like the wave to be. This is done by giving a value to the *amplitude modulator.*

WAVEFORMS:

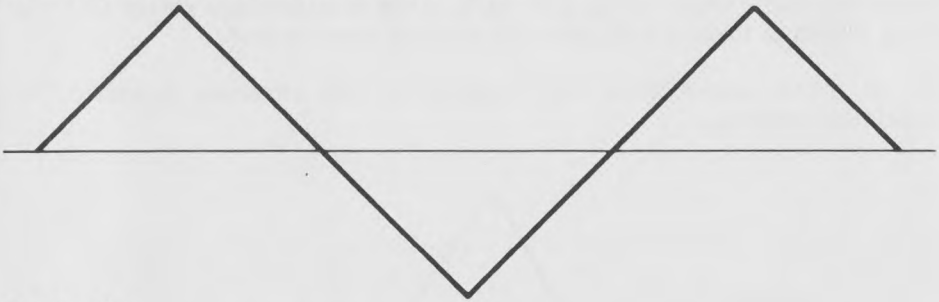
They couldn't have coined a better term for this feature of sound; a waveform is the form or shape of a wave. Waveforms dictate the more subtle aspects of sound, often referred to as the *timbre.*

The SID offers four basic waveforms. We list them below and offer some insight into their qualities:

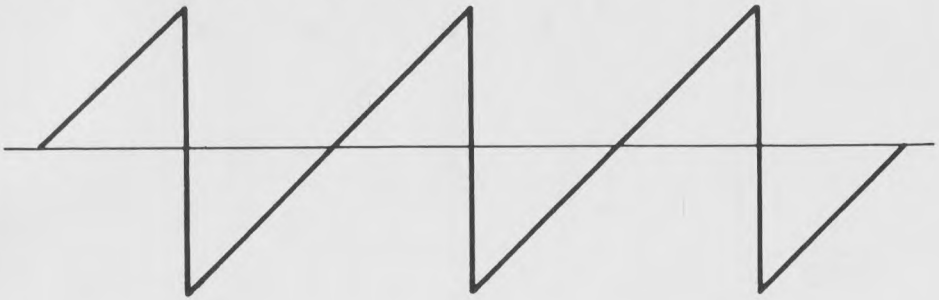
Rectangular: (Or Pulse)
A hollow reedy sound. Much like a clarinet.



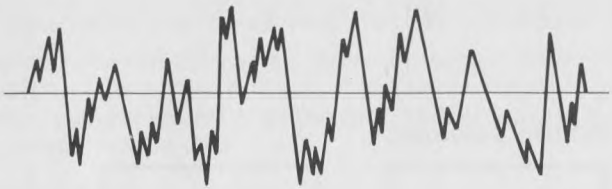
Triangular:
A clean, pure sound. Much like the flute.



Sawtooth:
A rough abrasive sound. Rich in harmonics.



White Noise:
Like a radio between channels. Explosion sounds.



ENVELOPES:
Attack Decay Sustain Release

The *envelope* of a sound defines the volume of a wave at each stage of its life. This cycle is divided into four stages, all of which are programmable.

In the *attack* stage, we tell SID how long it should take to get the wave from its beginning volume (0) to its maximum value. In the *decay* stage we tell SID how long it should take to drop from the maximum value to a lower level. In the *sustain* stage we tell SID what percentage of the maximum value the sound should drop to. Finally, in the *release* stage we tell SID how long it should take to drop from the sustain level to zero.

All of these values must be registered in the *envelope generator* for maximum control.

NOTES ON SID:

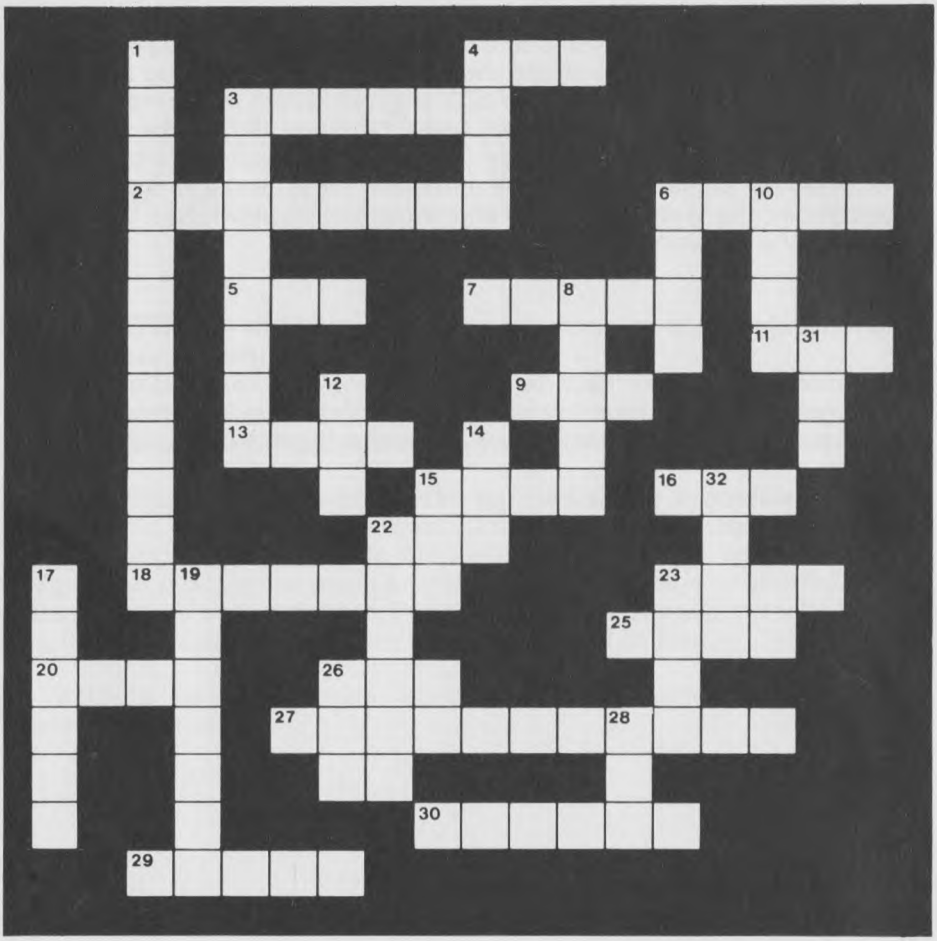
SID provides three independent voices. This means that three seperate sounds can be produced at the same time.

A good way to describe SID is to call it a group of cells or *registers.* There are 29 registers in all. We create the desired sound by *poking* a value into each register. Each voice within SID is given seven registers which are used to define the sound. Two of these registers define the pitch (tone oscillator), two define the volume (amplitude modulator), two define the attack, decay, sustain and release (envelope generator) and the remaining one defines the waveform (gate and waveform control). See the register listing in the appendix of this quide for details.

The remaining eight registers are used for overall SID output control.

Programming complex sounds and music can be very tedious without music generating programs. If you plan to program the SID often you might find that such programs make your music-making more enjoyable.

See the references in the back of this guide for more reading if you're interested in learning more.



ACROSS:

2. Key used to clear screen and return cursor to upper left.
3. COMMAND: Compares file in memory to file on storage device.
4. The smallest significant unit of data. (Always a 1 or a 0)
5. COMMAND: Returns the ASCII number for first character in a string.
6. COMMAND: Branches to a specified subroutine.
7. The most frequently used "combination" key.
9. Central Processing Unit (Abbr.)
11. COMMAND: Ends execution.
13. Mode in which program is altered.
15. COMMAND: Causes program to list.
16. Logical operator meaning "negative".
18. Number of available colors.
20. COMMAND: Used with "FOR" at the end of a loop.
22. COMMAND: Precedes a comment message in a program.
23. COMMAND: Reports contents of a memory location.
25. COMMAND: Continues program execution from line where stopped.
26. Type of memory used when programming.
27. The name of your computer.
29. COMMAND: Tells computer to output.
30. A group of characters.

For answers to crossword puzzle, fill out registration form.

DOWN:

1. Programmable Keys.
3. Holding cell for a specified value in BASIC.
4. One character in data terminology.
6. KEYWORD: Branches to specified line in program.
8. COMMAND: Prints message and waits for user to input string value.
10. COMMAND: Stores file in memory on an external device.
12. Sound making chip in 64 (abbr.)
14. COMMAND: Reserves space for an array.
15. COMMAND: Returns the number of characters in a string expression.
17. Format for instructing computer.
19. Whole number (without decimal point).
22. Change a filename.
23. COMMAND: Changes contents of specified memory location.
26. Type of memory that user cannot "write" to.
28. COMMAND: Tells computer to enter run-time mode.
31. COMMAND: Erases program in memory.
32. COMMAND: Begins channel with peripheral device.

RECOMMENDED READING

BOOKS

Commodore 64 Programmer's Reference Guide
Commodore Business Machines/Howard W. Sams
This reference guide is the best source of technical and programming information for the Commodore 64. Excellent source of information, can be hard to understand.

How to Program Your Commodore 64—BASIC for Beginners
HP Books
An excellent guide to BASIC programming for beginners. Also covers tape, disk drive, printer, sound and graphics. Well written, concise, easy. Some charts, few illustrations.

Commodore 64 Graphics & Sound Programming
TAB Books Inc.
An excellent sound and graphics book for beginners. Heavily illustrated, concise, helpful charts.

COMPUTE!'s First Book of Commodore 64
COMPUTE! Books
Explains Commodore 64 design and technical operation. Includes general information as well as utilities and program listings.

MAGAZINES:

COMPUTE! GAZETTE
COMPUTE! Publications, Greensboro, NC
An excellent monthly publication devoted only to Commodore computers. Nice programming tips for both beginners and advanced users. Helpful program listings. Resource listings. The best C-64 advertisments.

RUN
Wayne Green Inc, Petersboro, NH
An up and coming Commodore-only monthly which seems to be oriented towards a less-sophisticated user. Similar to COMPUTE! Gazzette in format and content.

Editors Note:
We believe that these publications are well designed but are not suggesting that they are the best on the market. Of the many publications that we have evaluated and used for our own information, these are some of our favorites at the time of this publication. Should you decide to buy or subscribe to any of these publications please take the time to evaluate them and competitive products first. Until you become more experienced we recommend that you consider readability the most important criteria.

LET US HEAR FROM YOU

It is our hope that you found the Training Kit informative, interesting and entertaining. If you have any comments or suggestions they would be most welcome.

Northwest Software • 621 Minna • San Francisco Ca • 94103

NORTHWEST SOFTWARE